What is claimed is:

1. A data dictionary comprising:

an inverse fault-tolerant decoder implemented for an error-correction code configured to transform a data vector into a plurality of predetermined index values;

combinational logic configured to combine pairs of said index values to form corresponding combined hash indices; and

data storage configured as a hash table referencing indexed data corresponding to said combined hash indices.

2. The data dictionary according to claim 1 wherein said data vector comprises a bit-attribute vector.

3. The data dictionary according to claim 1 wherein said inverse fault-tolerant decoder implements a reverse perfect error correction code.

4. The data dictionary according to claim 3 wherein said reverse perfect error correction code comprises a reverse Golay code.

5. The data dictionary according to claim 1 wherein said inverse fault tolerant decoder is further configured to identify said data vector as one of (i) a border vector type located at a border of a decoding sphere and (ii) a non-border vector type located interior to said decoding sphere.

6. The data dictionary according to claim 1 wherein said inverse fault-tolerant decoder is configured to:

identify said data vector as a border vector type,

define an offset of said data vector from a center of a decoding sphere of an error-

correction code implemented by said inverse fault-tolerant decoder; and

identify all possible offsets from adjacent decoding spheres of said error-correction code until said combinations fill in all bit positions corresponding to said data vector such that centers of said adjacent decoding spheres correspond to said index values.

7. The data dictionary according to claim 1 wherein said fault-tolerant decoder implements a reverse Golay code and is configured to:

identify said data vector as a non-border vector type;

identify an offset vector of said data vector from a center of a central index decoding sphere representing a specified offset distance;

identify centers of adjacent decoding spheres within said specified offset distance of said data vector; and

combines said centers of said adjacent decoding spheres with said center of said central index decoding sphere to form pairs of indexes.

8. A method of accessing a dictionary comprising the steps of:

transforming a data vector into a plurality of predetermined index values;

combining pairs of said index values to form corresponding combined hash indices; and

referencing indexed data stored in a hash table corresponding to said combined hash indices.

9. The method according to claim 8 wherein said data vector comprises a bit-attribute vector.

10. The method according to claim 8 wherein said transforming step implements a reverse perfect error correction code.

11. The method according to claim 10 wherein said reverse perfect error correction code comprises a reverse Golay code.

12. The method according to claim 8 wherein said transforming step further includes a step of identifying said data vector as one of (i) a border vector type located at a border of a decoding sphere and (ii) a non-border vector type located interior to said decoding sphere.

13. The method according to claim 8 wherein said transforming step further comprises the steps of:

identifying said data vector as a border vector type,

defining an offset of said data vector from a center of a decoding sphere of an error-correction code implemented by said inverse fault-tolerant decoder; and

identifying all possible offsets from adjacent decoding spheres of said error-correction code until said combinations fill in all bit positions corresponding to said data vector such that centers of said adjacent decoding spheres correspond to said index values.

14. The data dictionary according to claim 8 wherein said transforming step further comprises the steps of:

identifying said data vector as a non-border vector type;

identifying an offset vector of said data vector from a center of a central index decoding sphere representing a specified offset distance;

identifying centers of adjacent decoding spheres within said specified offset distance

of said data vector; and

combining said centers of said adjacent decoding spheres with said center of said central index decoding sphere to form pairs of indexes.

15. A data dictionary stored on a computer readable media, said data dictionary comprising:

inverse fault-tolerant decoder logic configured to transform a data vector into a plurality of predetermined index values;

combinational logic configured to combine pairs of said index values to form corresponding combined hash indices; and

a data storage structure configured as a hash table referencing indexed data corresponding to said combined hash indices.

16. The data dictionary according to claim 15 wherein said data vector comprises a bit-attribute vector.

17. The data dictionary according to claim 15 wherein said inverse fault-tolerant decoder implements a reverse Golay code.

18. The data dictionary according to claim 15 wherein said inverse fault tolerant decoder logic is further configured to identify said data vector as one of (i) a border vector type located at a border of a decoding sphere and (ii) a non-border vector type located interior to said decoding sphere.

19. The data dictionary according to claim 15 wherein said inverse fault-tolerant decoder logic is configured to:

identify said data vector as a border vector type,

define an offset of said data vector from a center of a decoding sphere of an error-correction code implemented by said inverse fault-tolerant decoder; and

identify all possible offsets from adjacent decoding spheres of said error-correction code until said combinations fill in all bit positions corresponding to said data vector such that centers of said adjacent decoding spheres correspond to said index values.

20. The data dictionary according to claim 15 wherein said fault-tolerant decoder logic implements a reverse Golay code and is configured to:

identify said data vector as a non-border vector type;

identify an offset vector of said data vector from a center of a central index decoding sphere representing a specified offset distance;

identify centers of adjacent decoding spheres within said specified offset distance of said data vector; and

combines said centers of said adjacent decoding spheres with said center of said central index decoding sphere to form pairs of indexes.